COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

DSFBA: Visualization

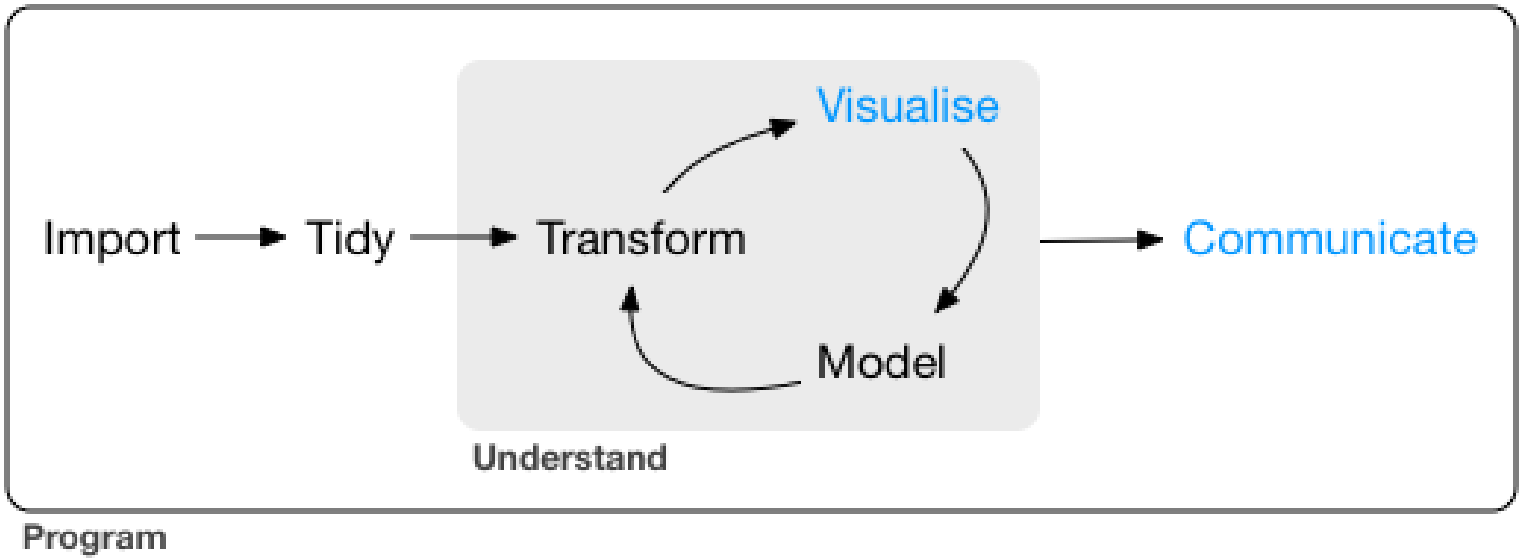
Data Science for Business Analytics

Thibault Vatter

Department of Statistics, Columbia University

10/27/2021

Today



Most of the material (e.g., the picture above) is borrowed from

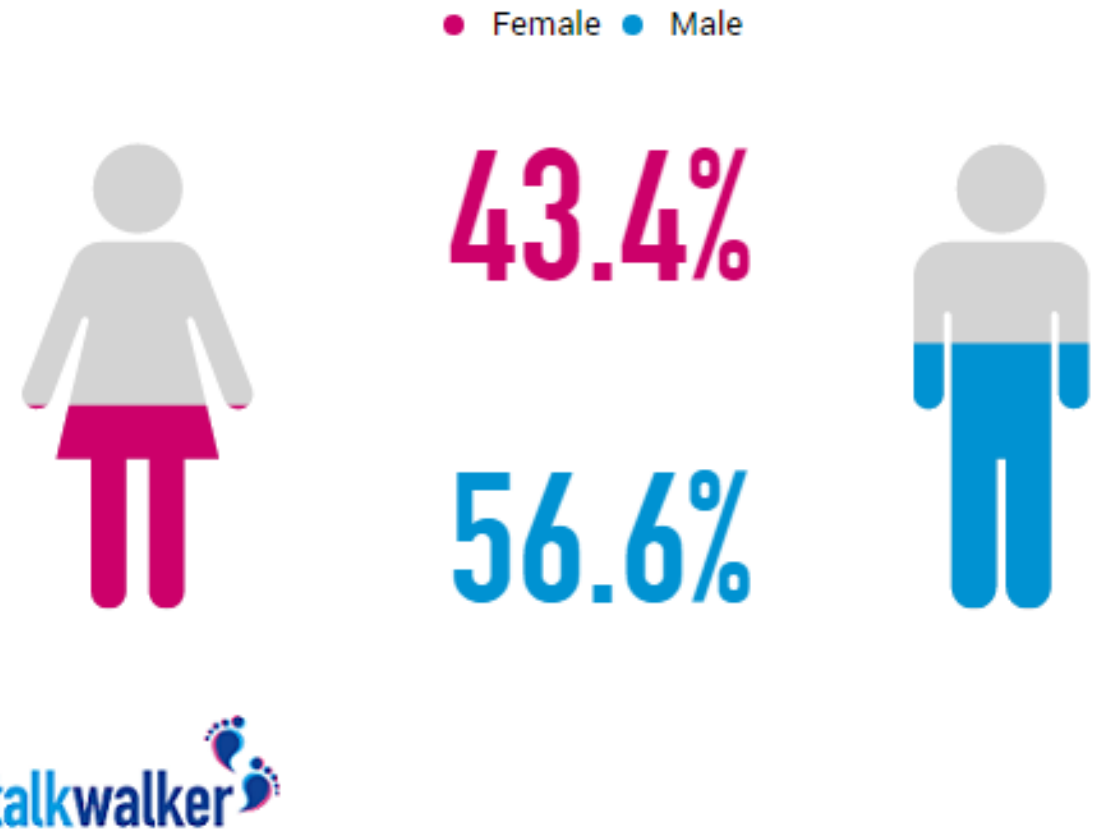
R for data science

Outline

Outline

Data content

- Makes no sense to use graphs for very small amounts of data.
- The human brain is capable of grasping a few values.



source: talkwalker.com

Data relevance

- Graphs are only as good as the data they display.
- No creativity can produce a good graph from poor data.

- Leinweber (author of *Nerds on Wall Street*):
  - The S&P500 could be "predicted"" at 75% by the butter production in Bangladesh.
  - ... Or 99% when adding cheese production in the USA, and the population of sheep.

Complexity

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

- Graphs shouldn't be more complex than the data they portray.
- Unnecessary complexity can be introduced by irrelevant
  - decoration
  - color
  - 3d effects
- . . . Collectively known as "chartjunk' '!

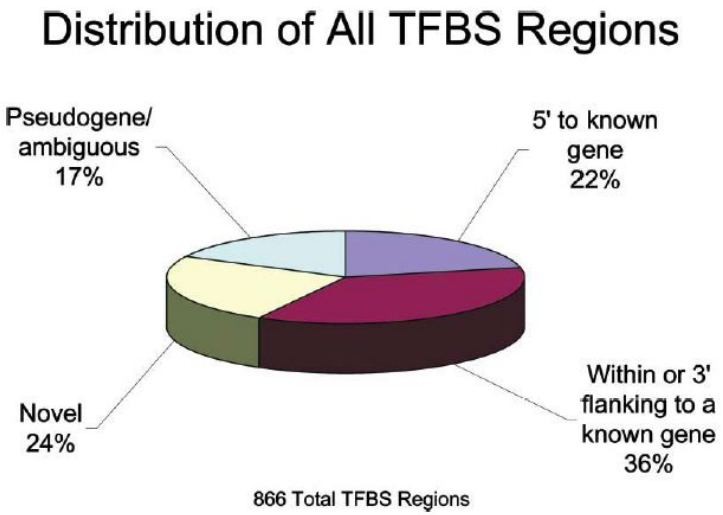


Figure 1. Classification of TFBS Regions

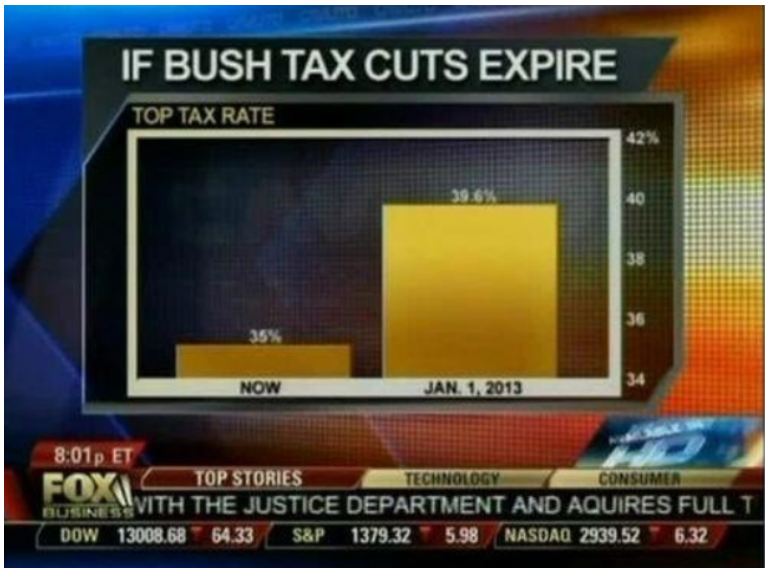
TFBS regions for Sp1, cMyc, and p53 were classified based upon proximity to annotations (RefSeq, Sanger hand-curated annotations, GenBank full-length mRNAs, and Ensembl predicted genes). The proximity was calculated from the center of each TFBS region. TFBS regions were classified as follows: within 5 kb of the 5′ most exon of a gene, within 5 kb of the 3′ terminal exon, or within a gene, novel or outside of any annotation, and pseudogene/ambiguous (TFBS overlapping or flanking pseudogene annotations, limited to chromosome 22, or TFBS regions falling into more than one of the above categories).

source: Cawley S, et al. (2004), Cell 116:499-509, Figure 1

Distorsion

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

- Graphs shouldn't be distorted pictures of the portrayed values:
  - Can be either deliberate or accidental.
  - Useful to know how to produce truth bending graphs.
  - Misleading often used as a synonym of distorted.
  - See https://en.wikipedia.org/wiki/Misleading_graph

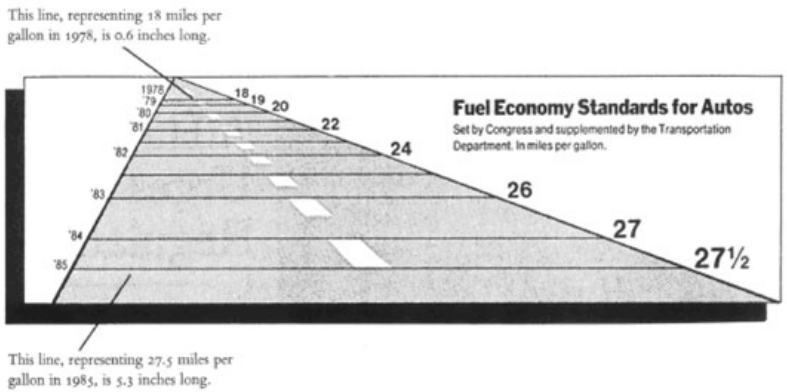


source: statisticshowto.com/misleading-graphs/

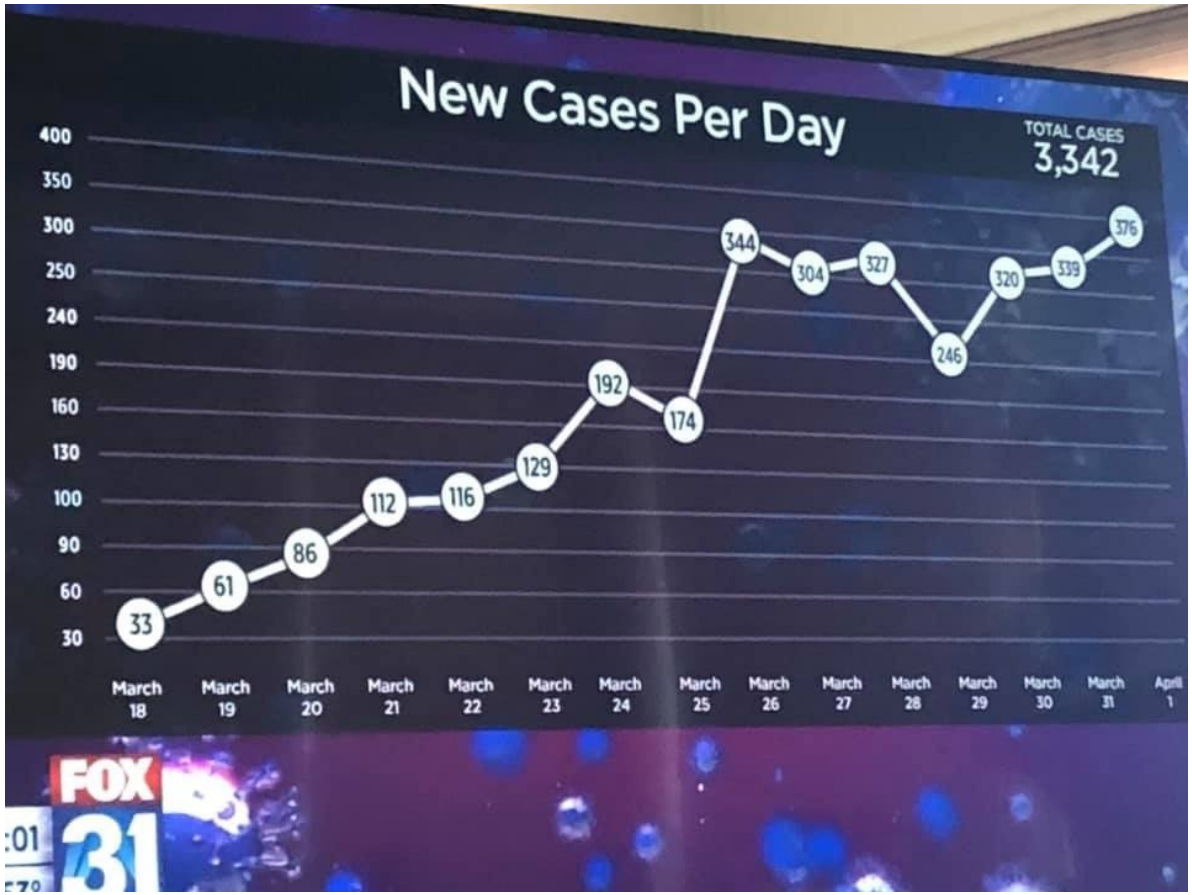
More on distorsion

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

- Common sources of distortion:
  - 3 dimensional "effects' '.
  - linear scaling when using area or volume to represent values.
- The "lie factor' ':
  - Measure of the amount of distortion in a graph.
    - lie factor $= \frac{\text{size of effect shown in graphic}}{\text{size of effect shown in data}}$
    - Don't take this too seriously, defined by Ed Tufte of Yale.
  - If > 1, the graph is exaggerating the effect.



$$\text{lie factor} = \frac{\frac{5.3-0.6}{0.6}}{\frac{27.5-18}{18}} = 14.8!$$


New Cases Per Day
TOTAL CASES
3,342
400
350
300
250
240
190
160
130
100
90
60
30
33
61
86
112
116
129
192
174
344
304
327
246
320
339
376
March 18
March 19
March 20
March 21
March 22
March 23
March 24
March 25
March 26
March 27
March 28
March 29
March 30
March 31
April 1
FOX
31

Drawing good graphs

- The three main rules:
  - If the "story' ' is simple, keep it simple.
  - If the "story' ' is complex, make it look simple.
  - Tell the truth – do not distort the data.
- Specifically:
  - There should be a high data to chart ratio.
  - Use the appropriate graph for the appropriate purpose.
    - Most graphs presented in Excel are POOR CHOICES!
    - In particular, never use a pie chart!
  - Make sure that the graph is complete:
    - All axes must be labeled.
    - The units should be indicated.
    - There should be a title.
    - A legend can provide needed additional information (e.g., for colors or line types).

A grammar of graphics

> *"A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the"scatterplot") and gain insight into the deep structure that underlies statistical graphics."* — *Hadley Wickham*

- `ggplot2` is an R implementation of the concept:
  - A coherent system for describing and creating graphs.
  - Based on The Grammar of Graphics.
  - Learn one system and apply it in many places.
  - The equivalent of `dplyr` for graphs.
- To learn more, read The Layered Grammar of Graphics.
- Implementations exist in other languages (e.g., Python)

The mpg data frame

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

- Data from the US EPA on 38 models of car:

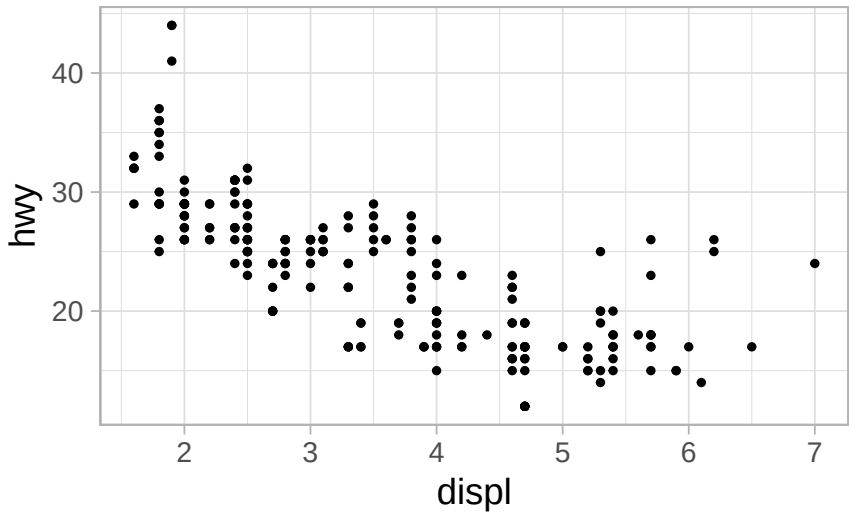
```
mpg %>% print(n = 5)
#> # A tibble: 234 x 11
#>   manufacturer model displ  year   cyl trans  drv     cty   hwy fl
#>   <chr>        <chr> <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
#> 1 audi         a4      1.8  1999     4 auto(~ f        18    29 p
#> 2 audi         a4      1.8  1999     4 manua~ f        21    29 p
#> 3 audi         a4      2    2008     4 manua~ f        20    31 p
#> 4 audi         a4      2    2008     4 auto(~ f        21    30 p
#> 5 audi         a4      2.8  1999     6 auto(~ f        16    26 p
#> # ... with 229 more rows, and 1 more variable: class <chr>
```

- Among the variables in mpg are:
  - displ, a car's engine size, in litres.
  - hwy, a car's fuel efficiency on the highway (in miles per gallon).
- A few questions
  - Do cars with big engines use more fuel ?
  - What does the relationship between engine size and fuel efficiency look like? Positive? Negative? Linear? Nonlinear?

Creating a plot

- A simple scatterplot:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



- A graphing template:

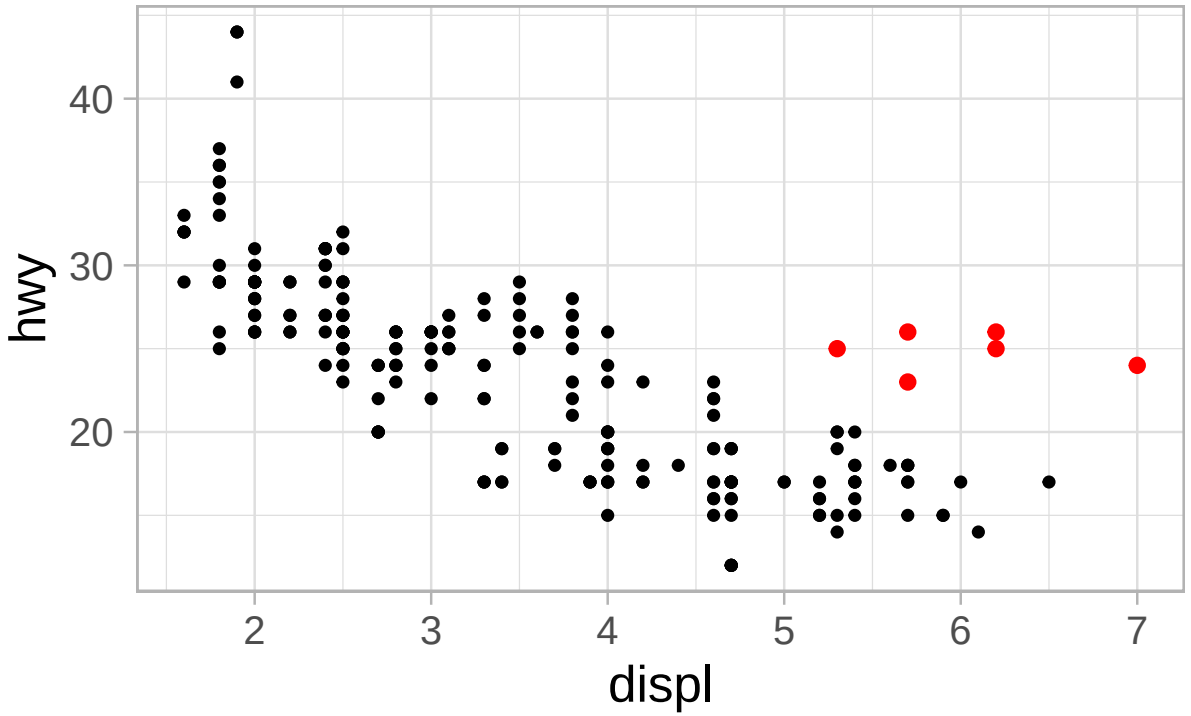
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Outline

Aesthetic mappings

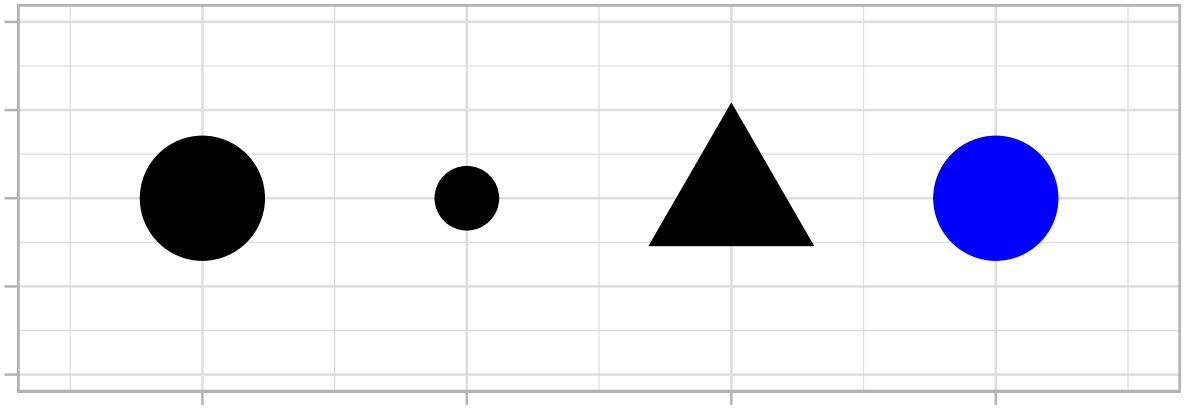
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

"The greatest value of a picture is when it forces us to notice what we never expected to see." — John Tukey



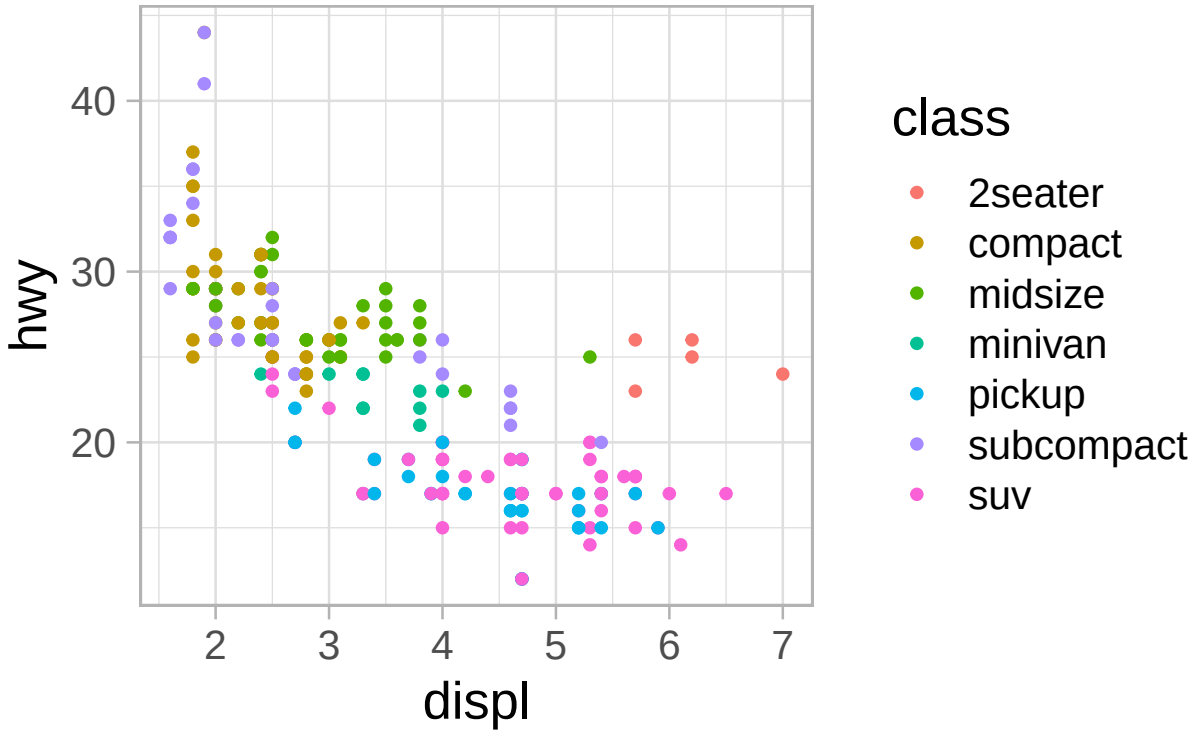
Aesthetic

- How to add a more variables to a two dimensional plot?
- By mapping them to an **aesthetic**:
  - A visual property of the objects in your plot.
  - Include the size, the shape, or the color of the points.
- We use the words
  - **"value"** to describe data,
  - and **"level"** to describe aesthetic properties.



Adding classes to your plot


```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

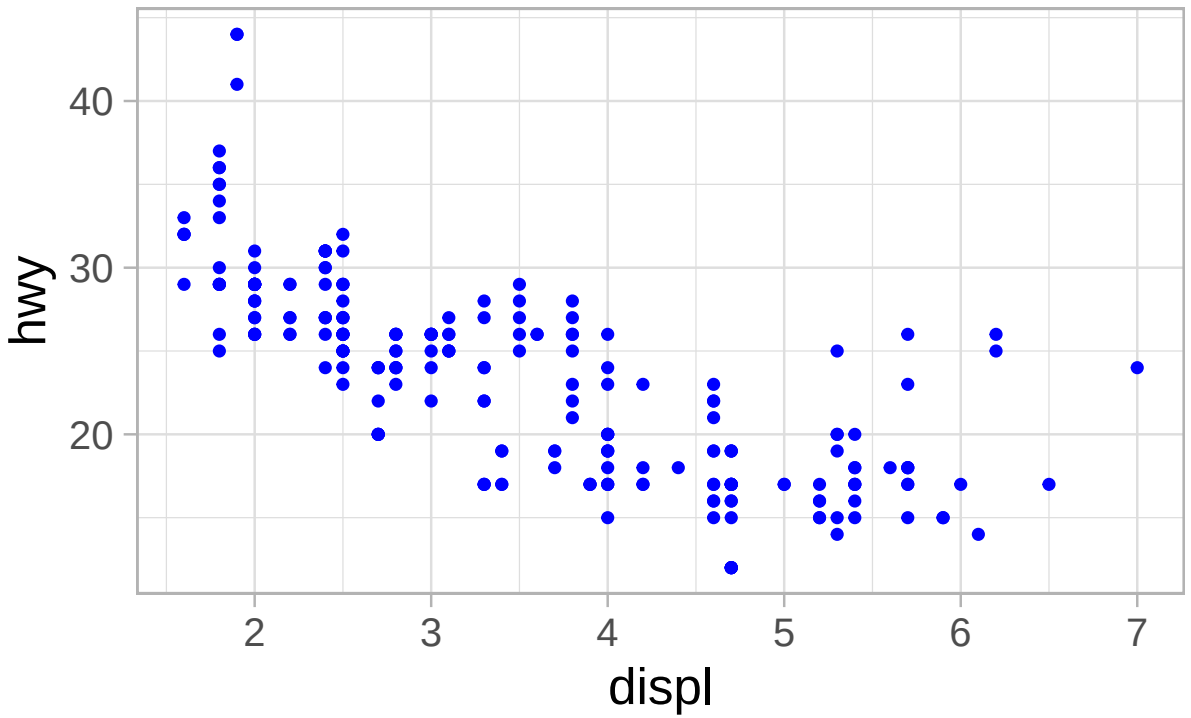


- If you prefer British English, use `color` instead of `color`.

Set the aesthetics manually

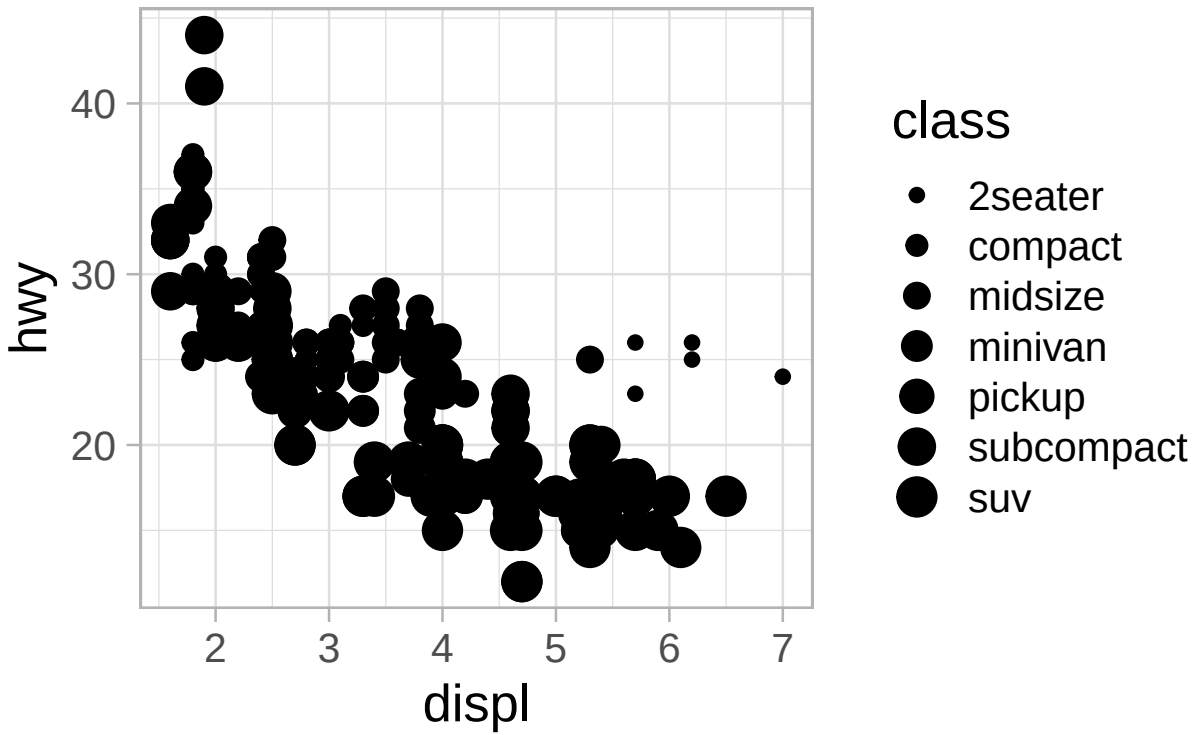
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```



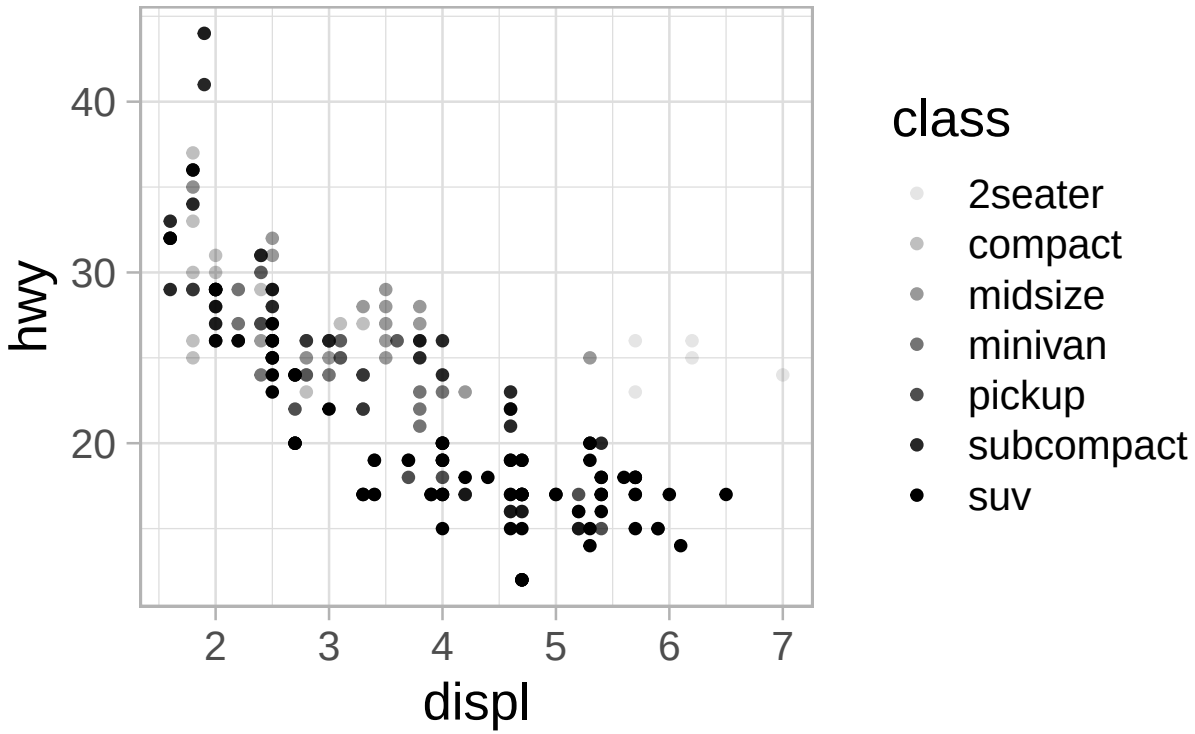
The size aesthetic


```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
#> Warning: Using size for a discrete variable is not advised.
```



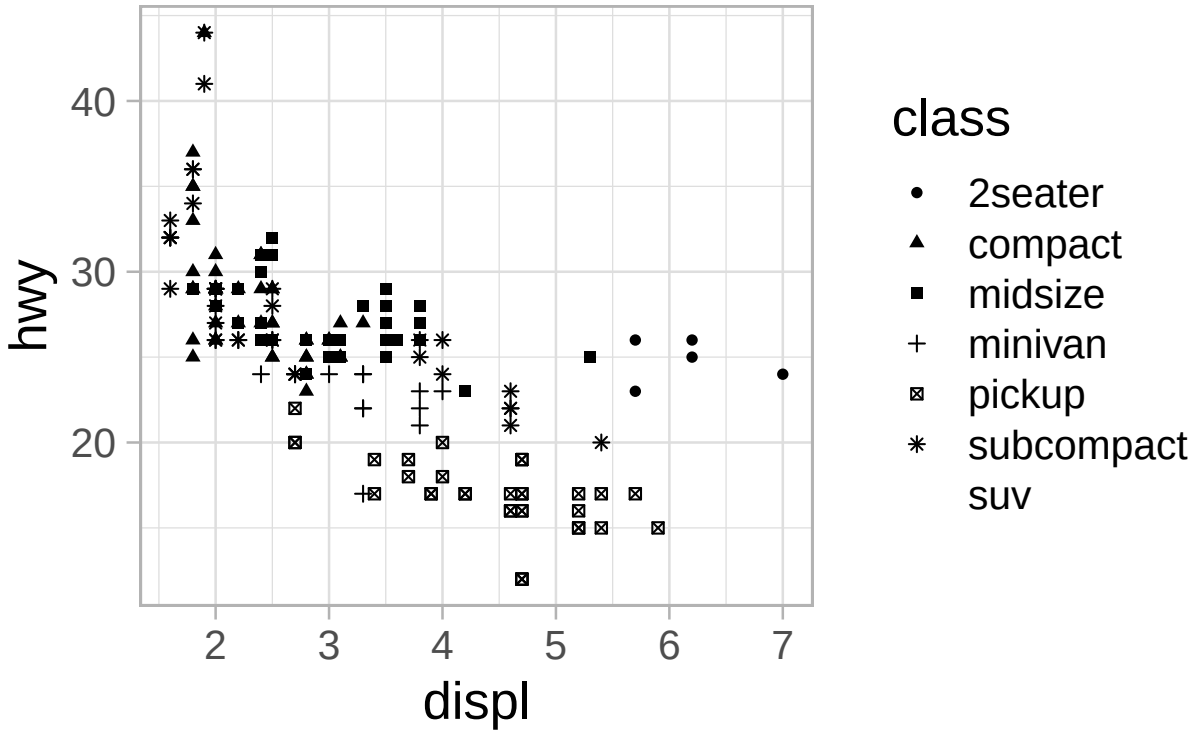
The alpha aesthetic


```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
#> Warning: Using alpha for a discrete variable is not advised.
```



The shape aesthetic


```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
#> Warning: The shape palette can deal with a maximum of 6 discrete
#> values because more than 6 becomes difficult to discriminate;
#> you have 7. Consider specifying shapes manually if you must
#> have them.
#> Warning: Removed 62 rows containing missing values (geom_point).
```



A remark on aesthetic values

- Need values that make sense for that aesthetic:
  - The name of a color as a character string.
  - The size of a point in mm.
  - The shape of a point as a number.

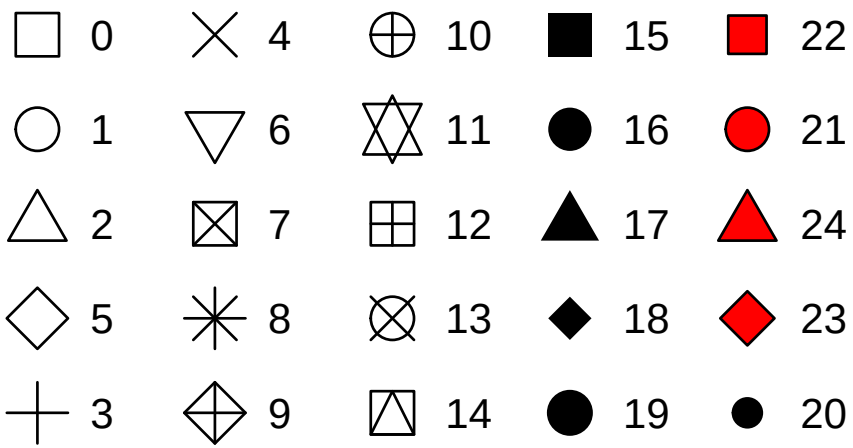
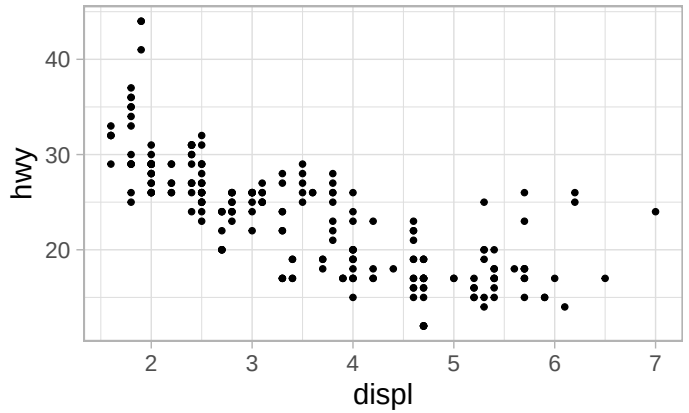


Figure 1: The hollow shapes (0–14) have a border determined by ‘color‘; the solid shapes (15–18) are filled with ‘color‘; the filled shapes (21–24) have a border of ‘color‘ and are filled with ‘fill‘.

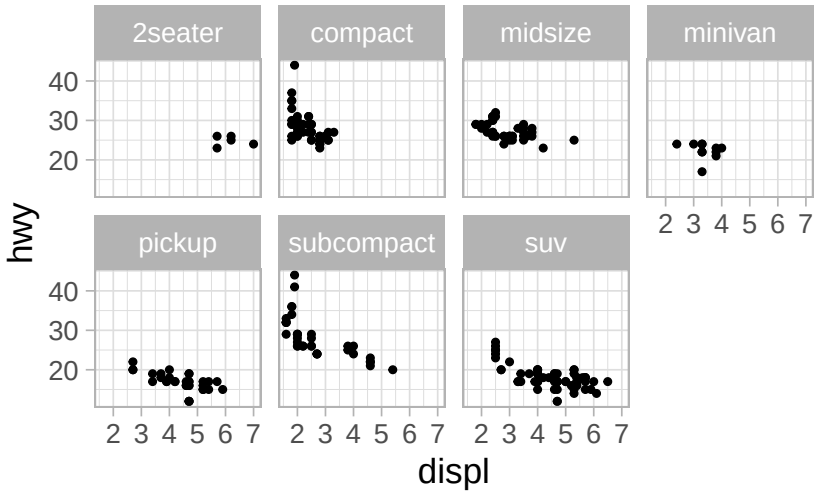
Facetting


```
(p <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)))
```



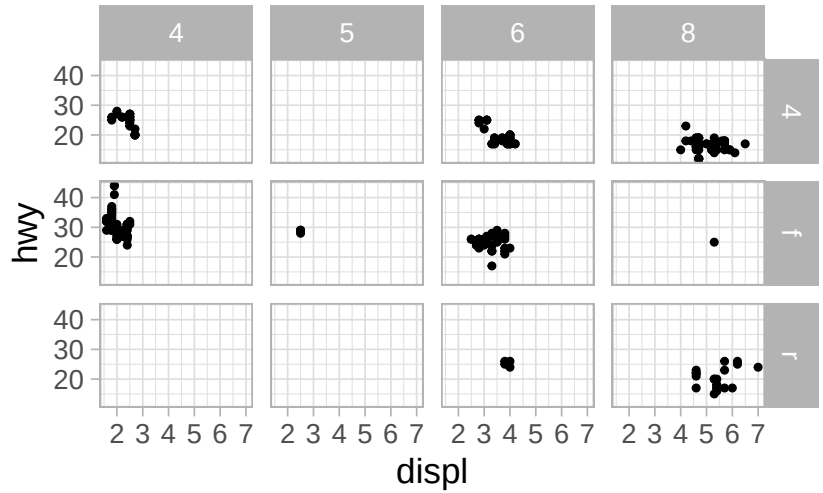
- Facet wrap

```
p + facet_wrap(~ class, nrow = 2)
```



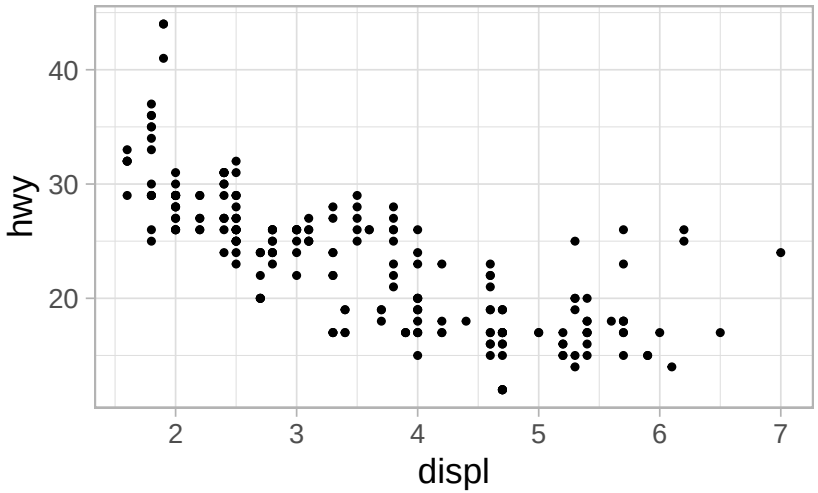
- Facet grid

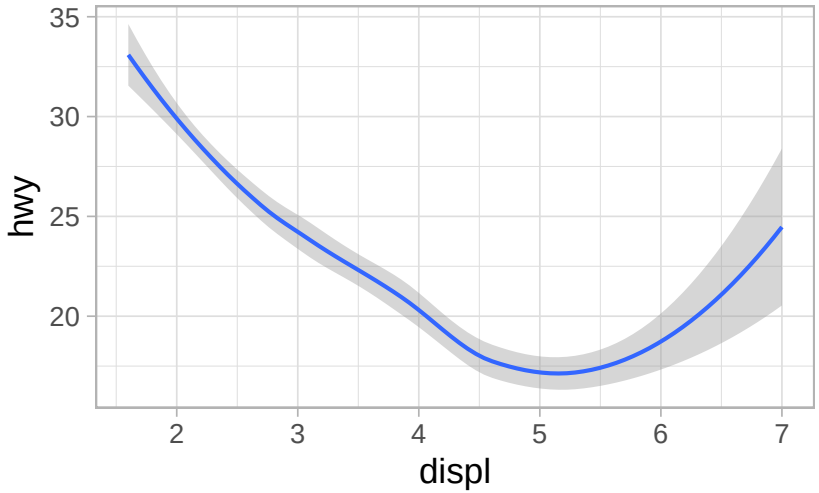
```
p + facet_grid(drv ~ cyl)
```



Outline

How are these two plots similar?





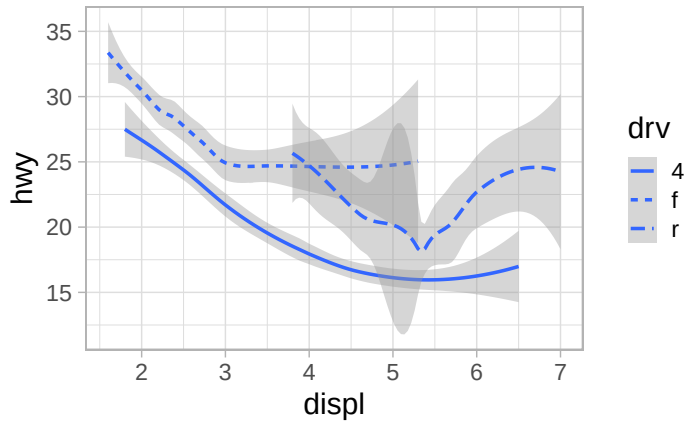
- A **geom**:
  - The object that a plot uses to represent data.
  - Plots often describeds by the geom type:
    - Bar charts use bar geoms.
    - Line charts use line geoms.
    - Boxplots use boxplot geoms.
  - An exception:
    - Scatterplots use the point geom.

Geometric objects

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

- Every **geom** function takes a `mapping` argument.
- But **not every aesthetic works with every geom**:
  - **shape** exists for `geom_point` but not for `geom_line`,
  - and conversely for **linetype**.

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv))
```

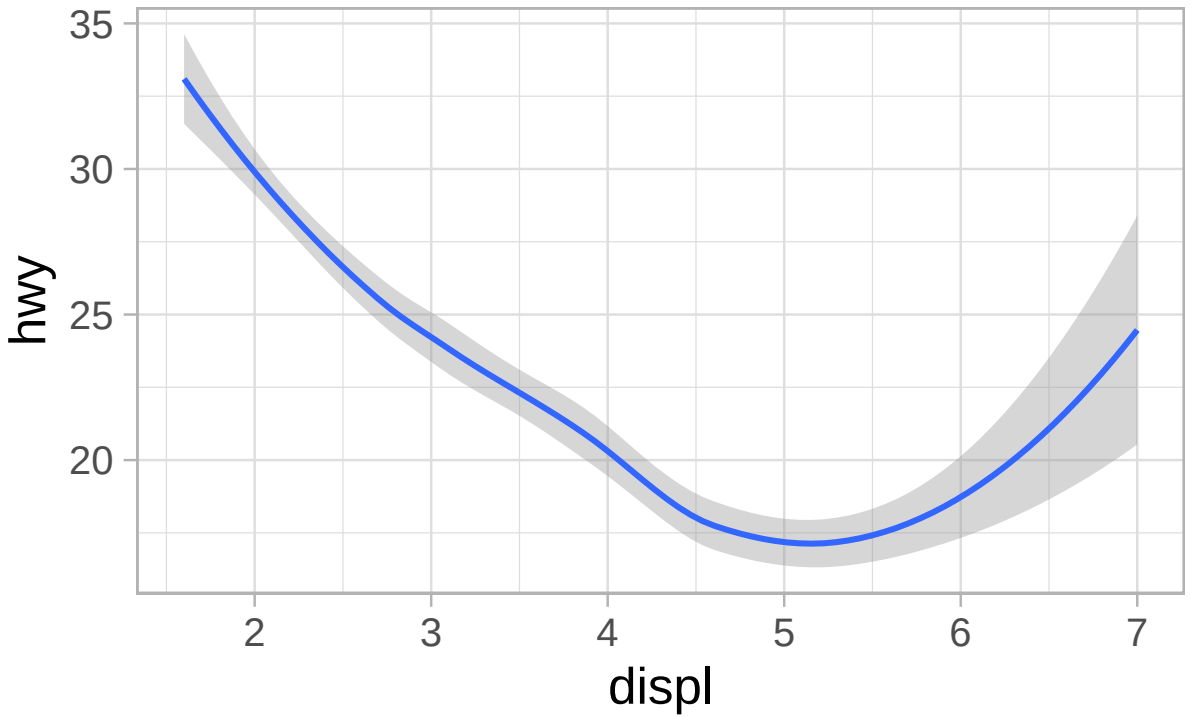


- Additionally
  - `ggplot2` provides over 30 geoms.
  - extension packages provide even more.
  - Use RStudio's data visualization cheatsheet.
  - To learn more about any single geom, use help: `?geom_smooth`.

Geoms and legends

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

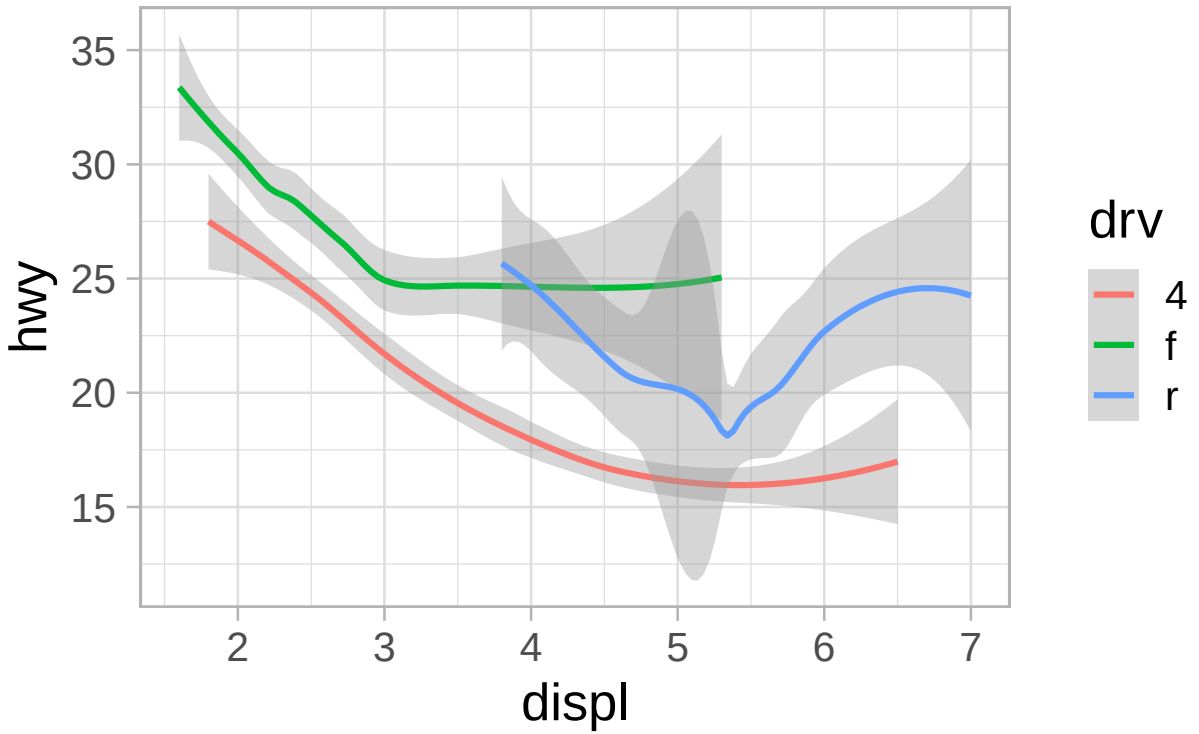
```
ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy))
```



Geoms and legends

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

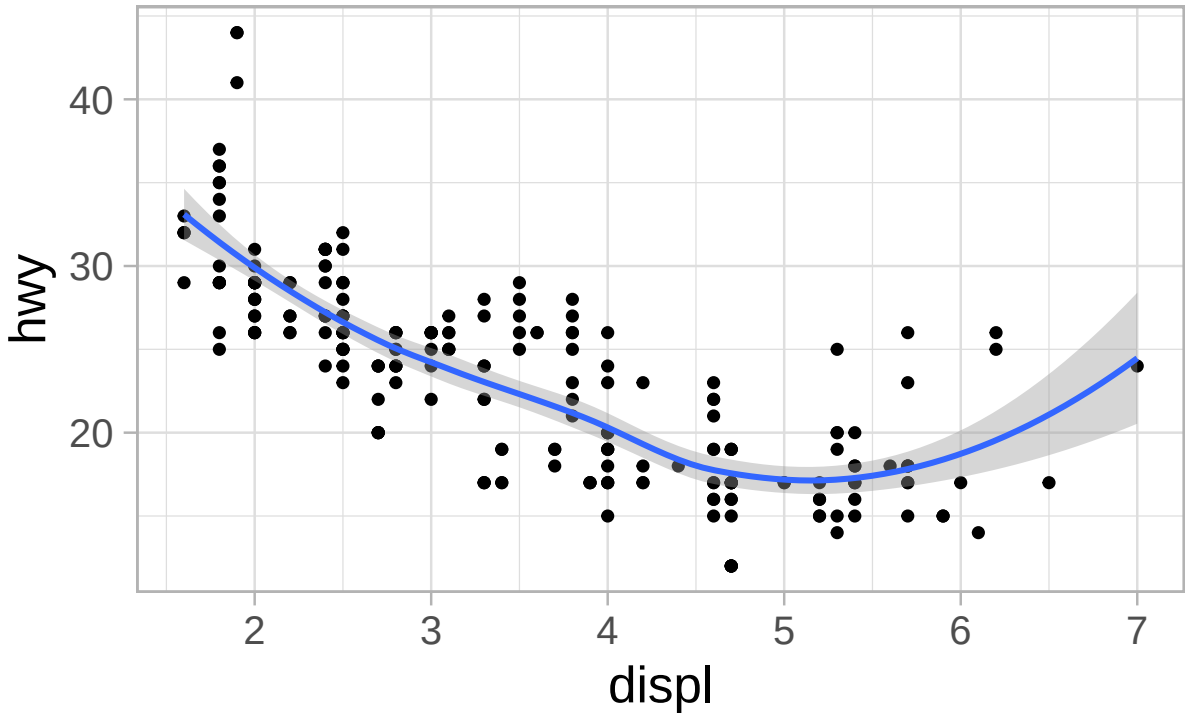
```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, color = drv))
```



Multiple geoms in the same plot

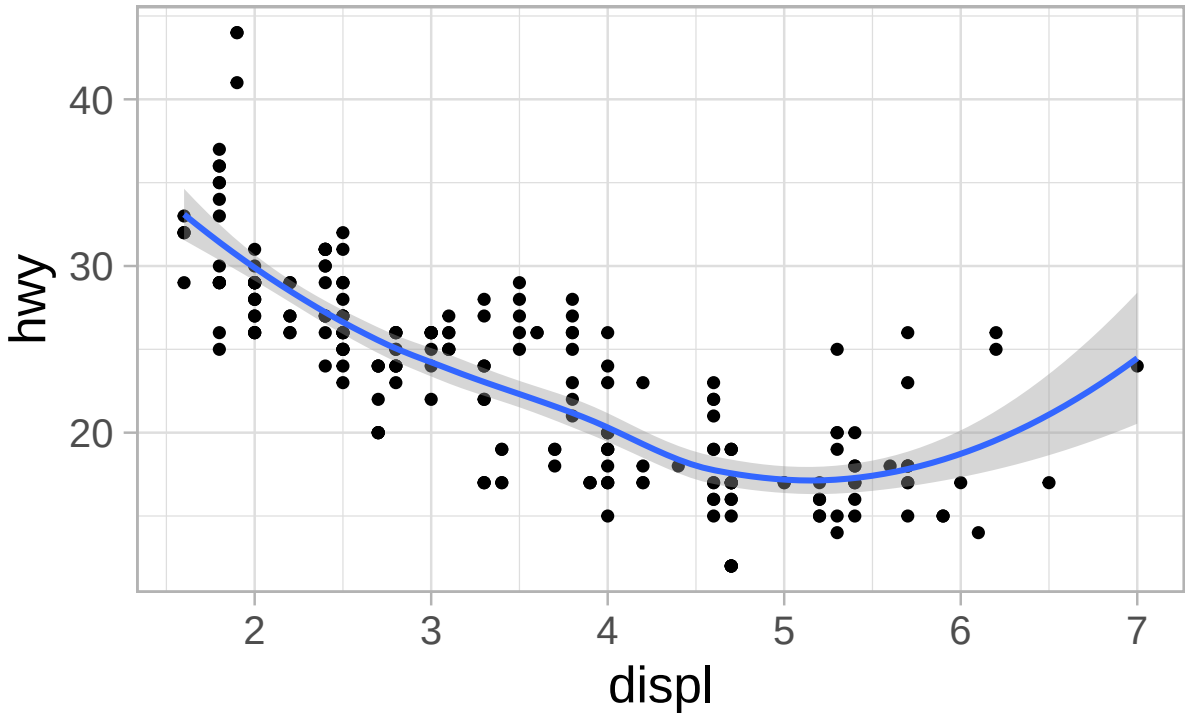
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



A better way

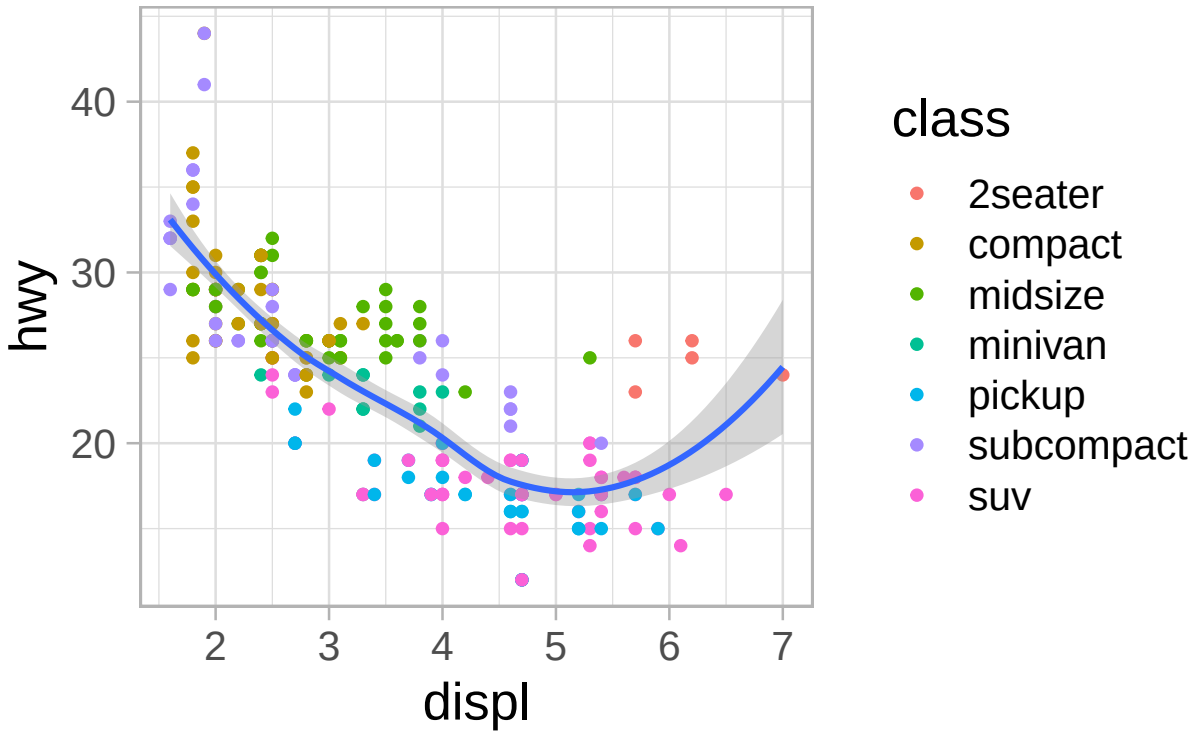

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```



Local vs global mappings

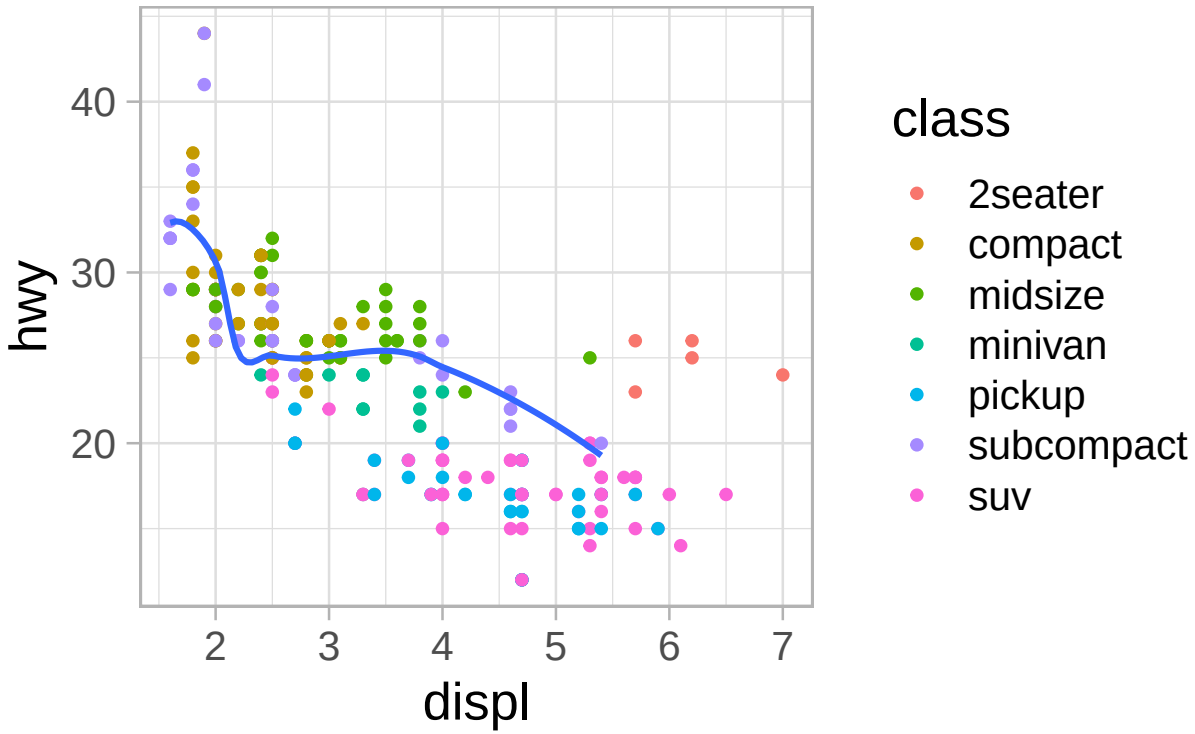
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth()
```



Layer dependent `data`

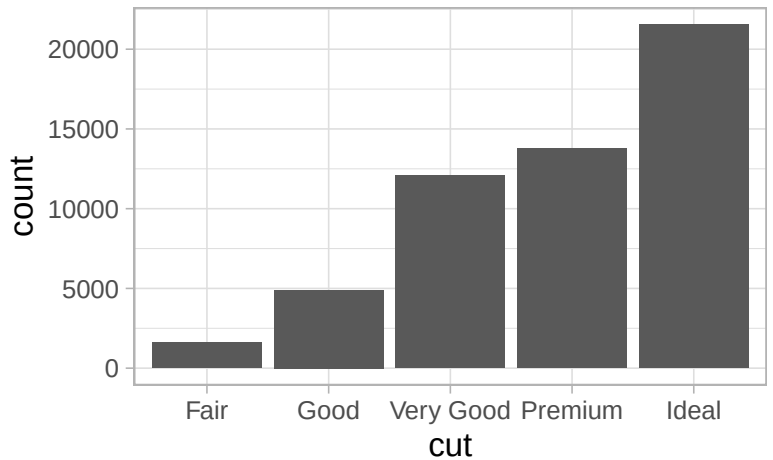

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```



Beyond scatterplots

- Other graphs, like bar charts, calculate new values to plot.
  - Bar charts, histograms, and frequency polygons:
    - Bin data.
    - Plot bin counts (number of points falling in each bin).
  - Smoothers:
    - Fit a model to your data.
    - Plot predictions from the model.
  - Boxplots:
    - Compute a robust summary of the distribution.
    - Display a specially formatted box.

```
ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut))
```



Statistical transformations

- A **stat**:
  - The algorithm used to calculate new values for a graph.
  - Short for statistical transformation.

1. **`geom_bar()`** begins with the **`diamonds`** data set

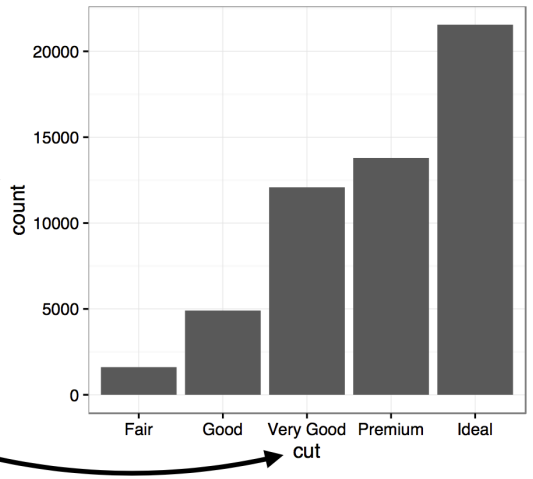
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

2. **`geom_bar()`** transforms the data with the "count" stat, which returns a data set of cut values and counts.

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

3. **`geom_bar()`** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.



- `ggplot2` provides over 20 stats.
- Each stat is a function, get help as usual, e.g. `?stat_bin`.
- Use RStudio's data visualization cheatsheet for a complete list.

Geom and stat

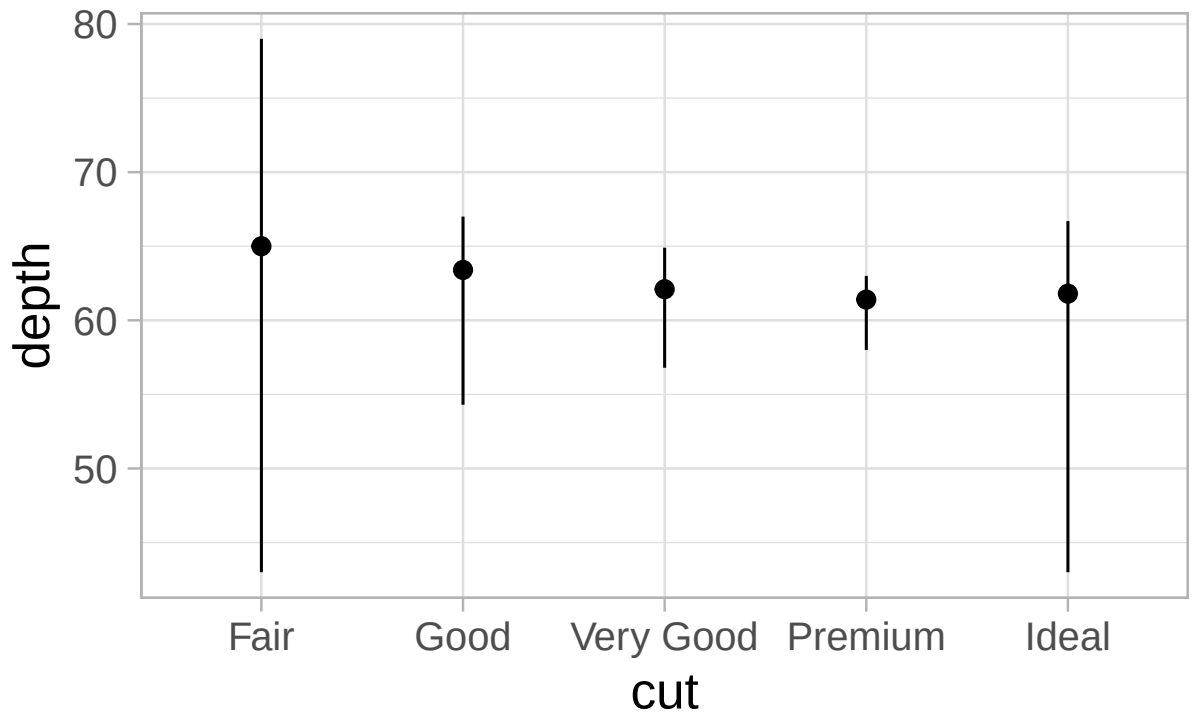
- Every geom has a default stat and conversely.
  - `?geom_bar` shows that the default value for `stat` is "count".
  - Means that `geom_bar()` uses `stat_count()`.
  - `?stat_count` has a section called "Computed variables" with two new variables: `count` and `prop`.
- You can generally use geoms and stats interchangeably!

```
ggplot(data = diamonds) +
  stat_count(mapping = aes(x = cut))
```

- Typically, use geoms without worrying about the stat.
- Three reasons to use a stat explicitly:
  - To override the default stat.
  - To override the default mapping from transformed variables to aesthetics.
  - To draw greater attention to the stat in your code.

Use a stat explicitely I

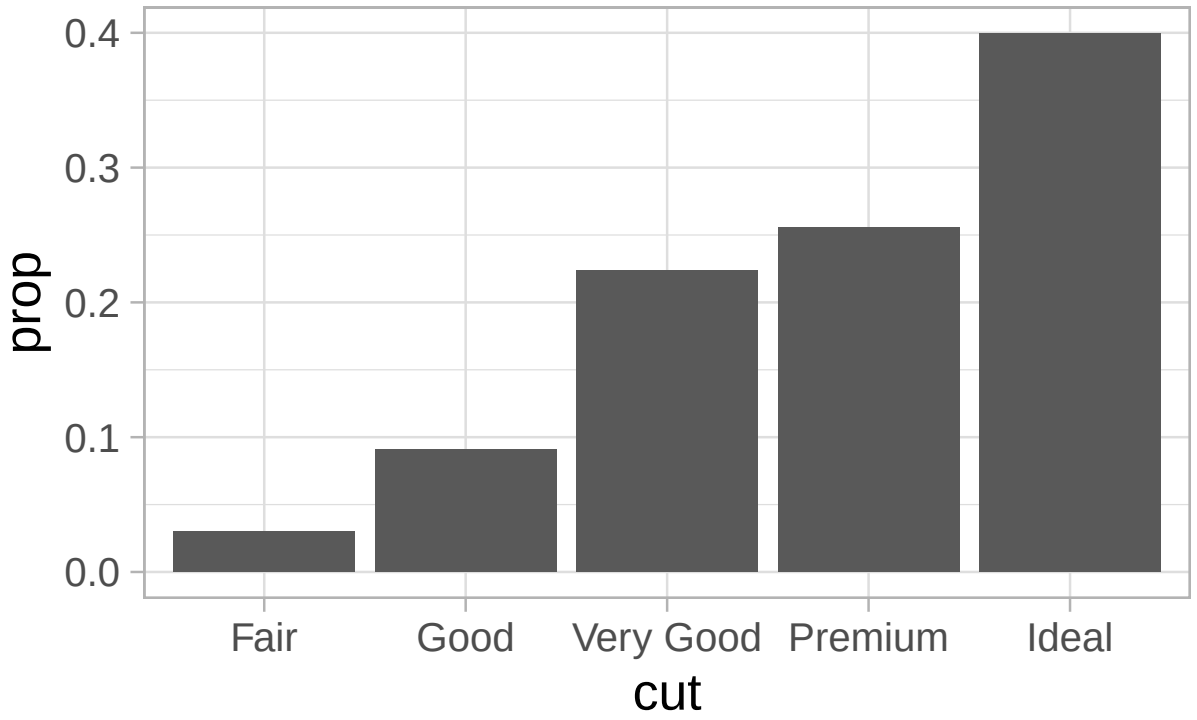

```
ggplot(data = diamonds) +
      stat_summary(
        mapping = aes(x = cut, y = depth),
        fun.min = min,
        fun.max = max,
        fun = median
        )
```



Use a stat explicitely II

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

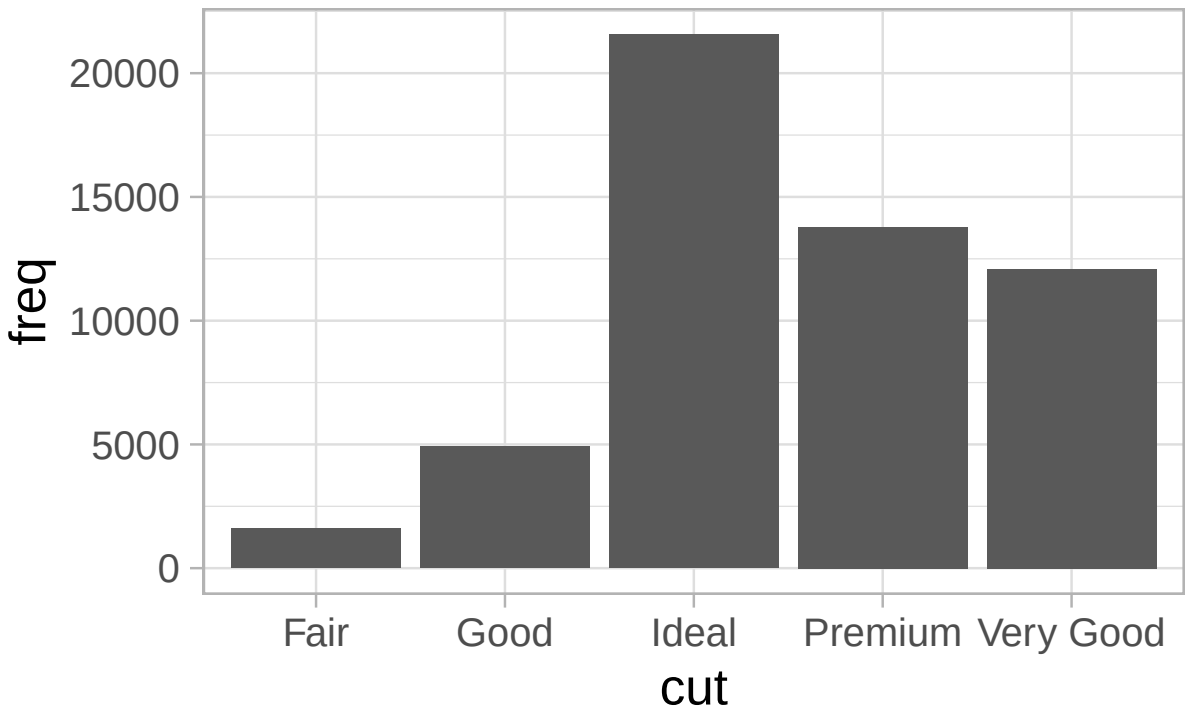
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop.., group = 1))
```



Use a stat explicitely III

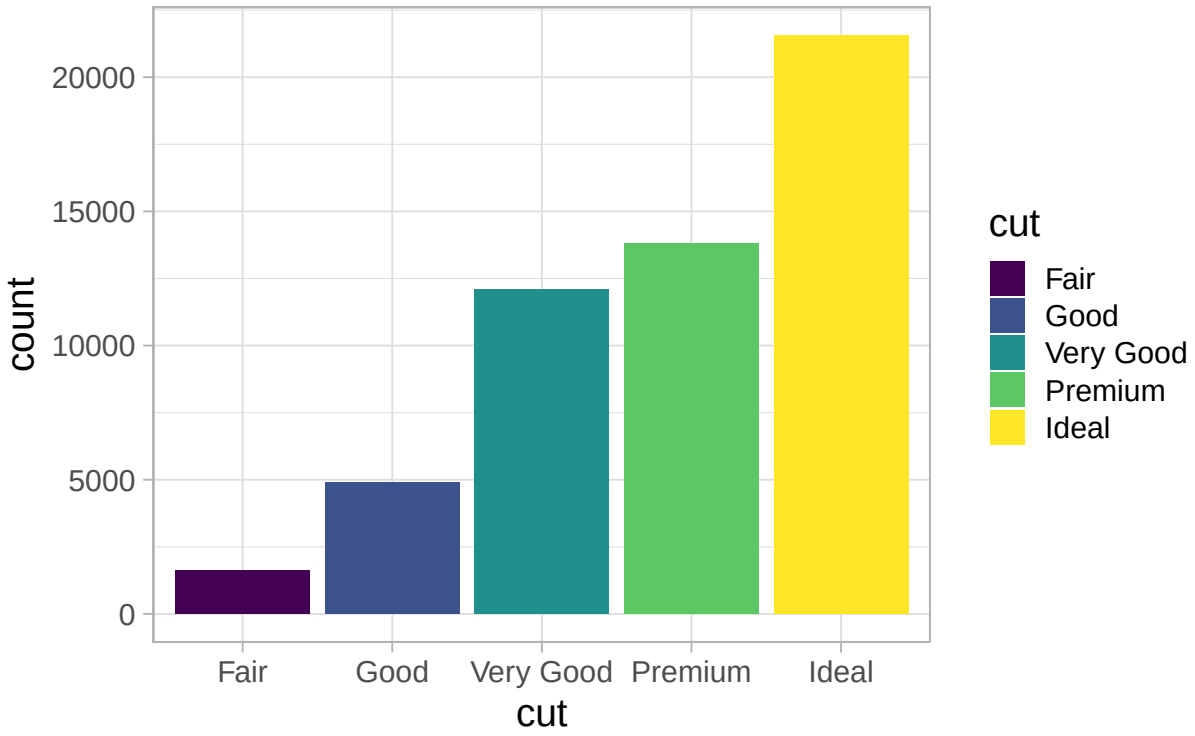
COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

```
demo <- tribble(~cut,          ~freq,
                "Fair",        1610,
                "Good",        4906,
                "Very Good",   12082,
                "Premium",     13791,
                "Ideal",       21551)
 ggplot(data = demo) +
   geom_bar(mapping = aes(x = cut, y = freq), stat = "identity")
```



The fill aesthetic

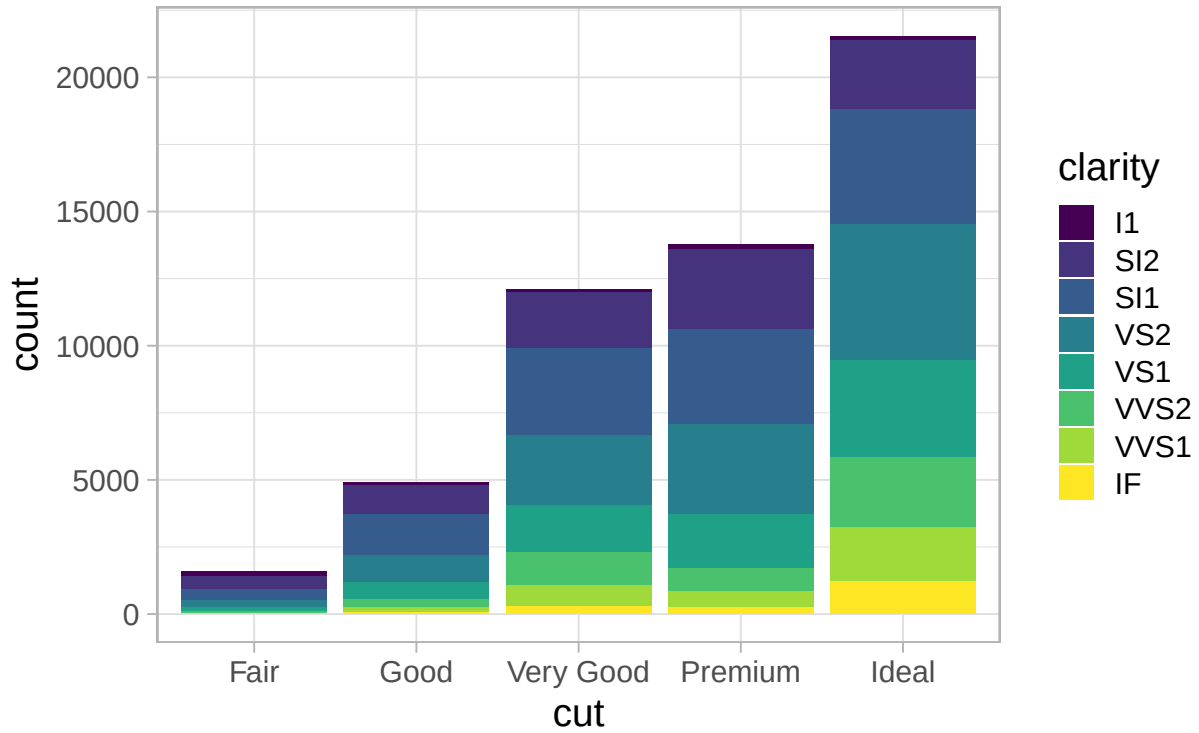

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```



Fill and position ajustements

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

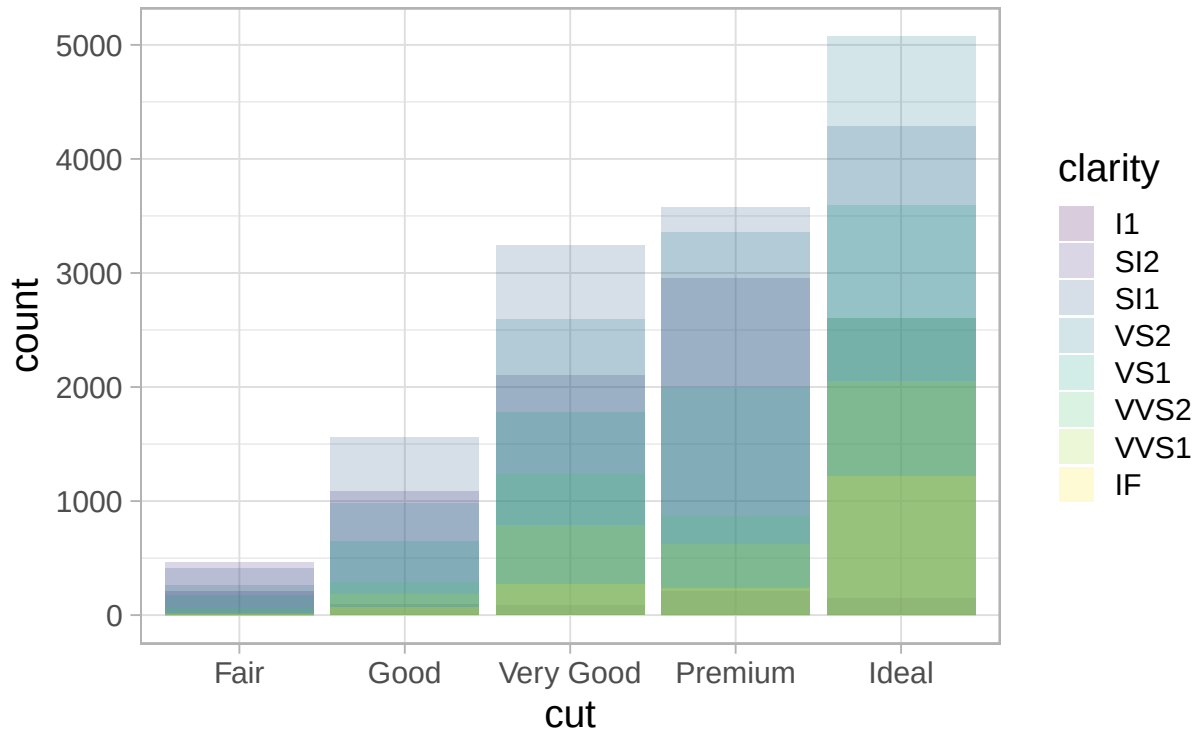
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity))
```



- Automatically stacked by the **position adjustement**.
- `?position_stack` to learn more.

Fill with `position = "identity"`

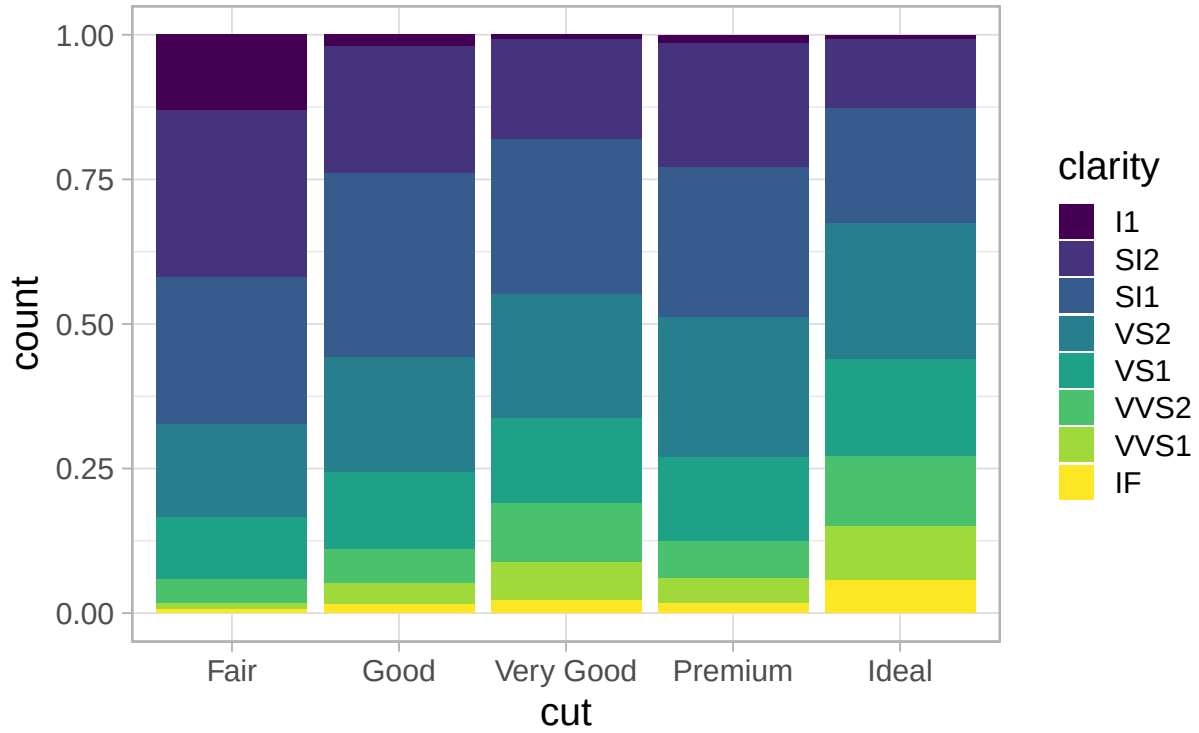

```
ggplot(data = diamonds, mapping = aes(x = cut, fill = clarity)) +
  geom_bar(alpha = 1/5, position = "identity")
```



- Not very useful for bars because of overlap.
- `?position_identity` to learn more.

Fill with `position = "fill"`

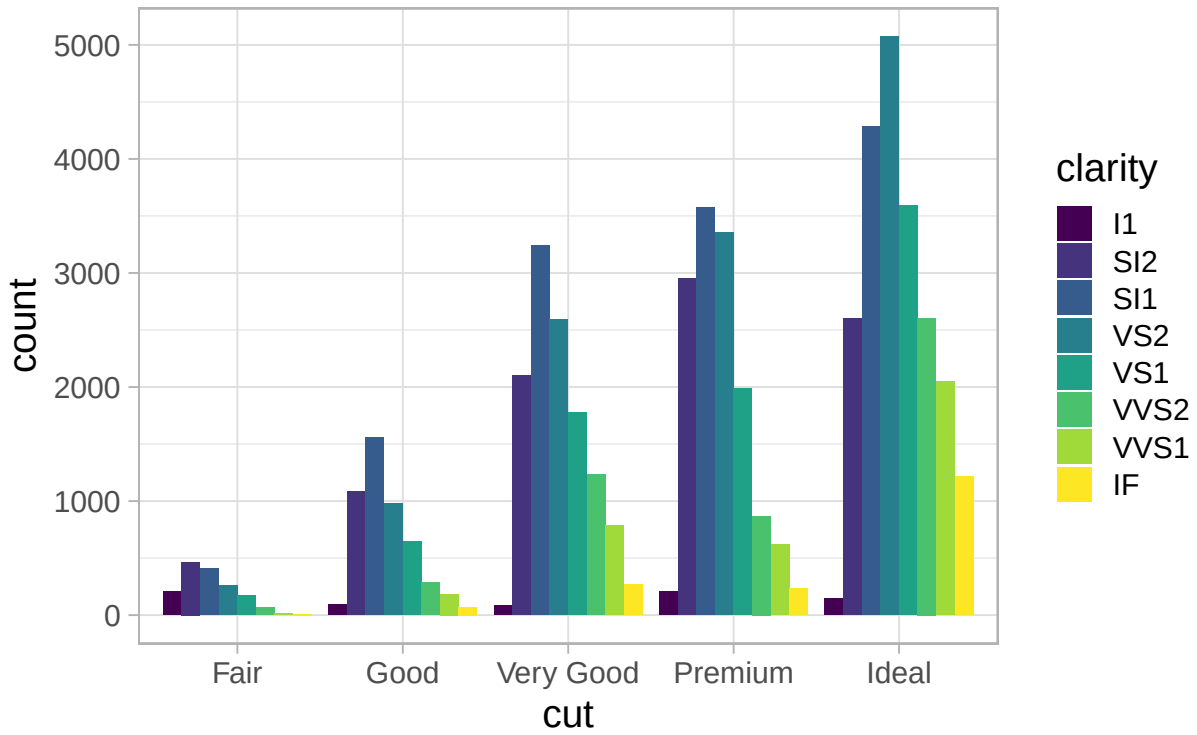

```
ggplot(data = diamonds) +
      geom_bar(mapping = aes(x = cut, fill = clarity), position = "fill")
```



- Makes it easier to compare proportions across groups.
- `?position_fill` to learn more.

Fill with `position = "dodge"`

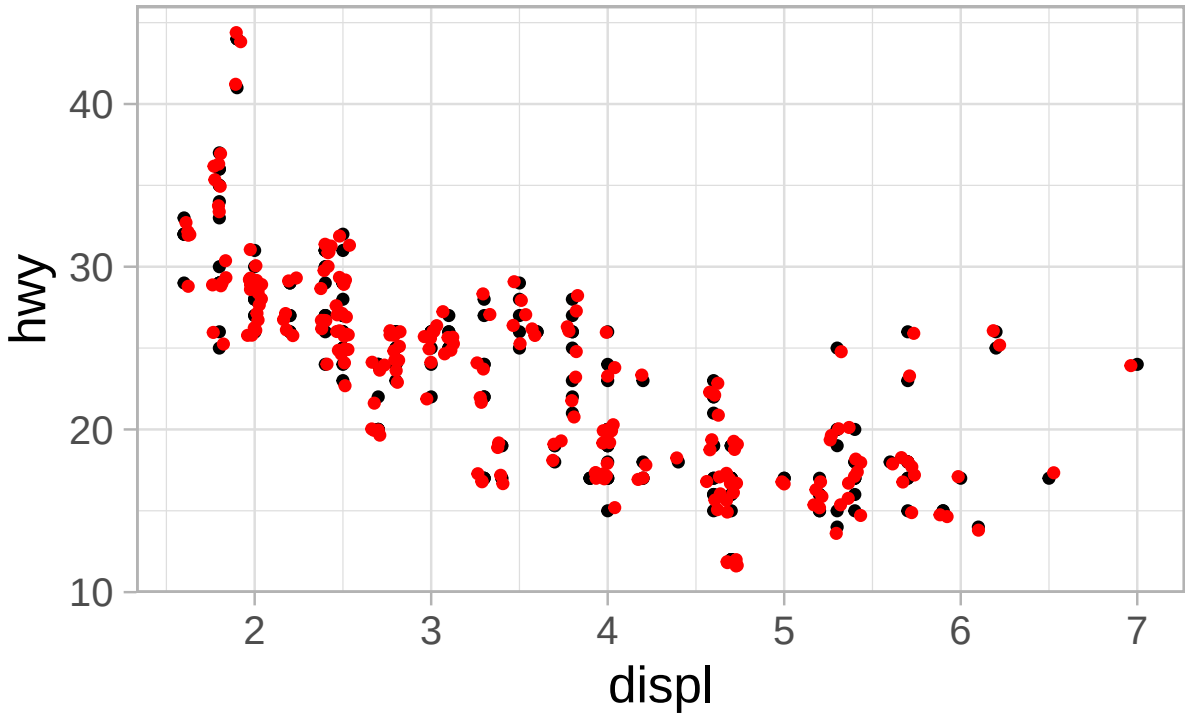

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge")
```



- Makes it easier to compare individual values.
- `?position_dodge` to learn more.

position = "jitter"


```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_point(position = "jitter", color = "red")
```



- Graph less/**more** accurate/**revealing** at small/**large** scales.
- ?position_jitter to learn more.

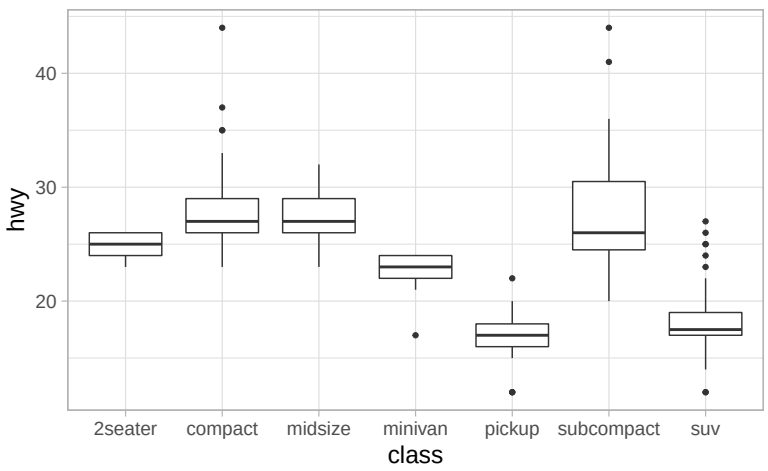
Outline

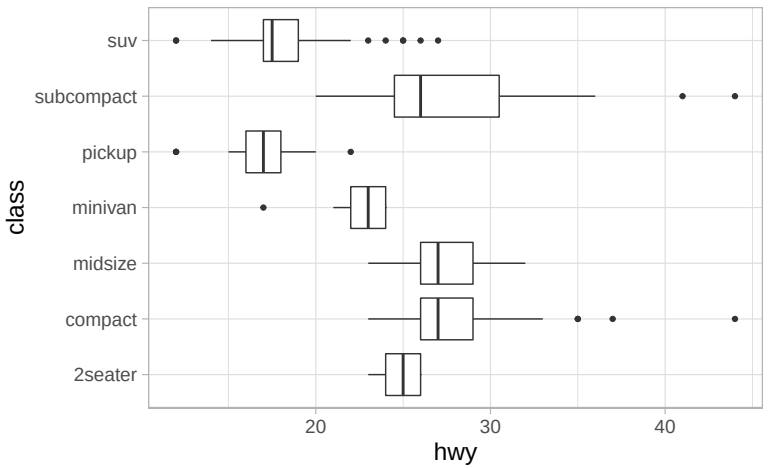
Coordinate systems

- The most complicated part of `ggplot2`.
- Default: the Cartesian coordinate system.
- Other systems occasionally helpful:
  - `coord_flip()` switches the x and y axes.
  - `coord_quickmap()` sets the aspect ratio correctly for maps.
  - `coord_polar()` uses polar coordinates.

coord_flip()


```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot() +
  coord_flip()
```



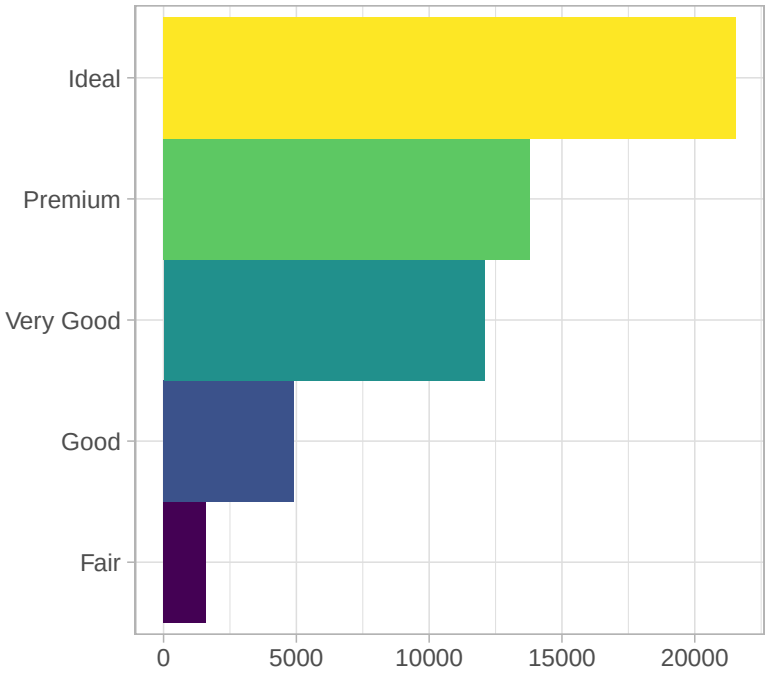


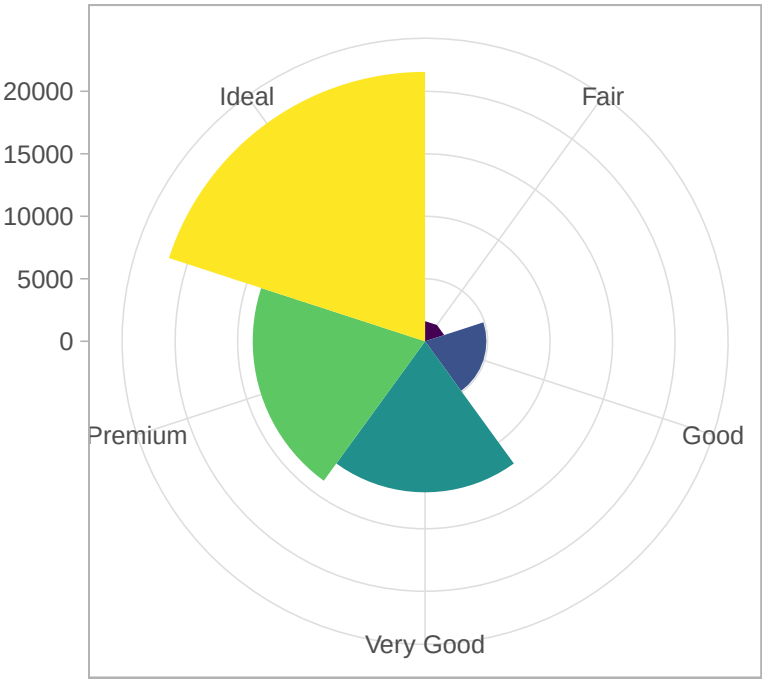
- Useful for:
  - horizontal boxplots,
  - and long labels.

coord_polar()

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

```
bar <- ggplot(data = diamonds) +
      geom_bar(mapping = aes(x = cut, fill = cut),
               show.legend = FALSE, width = 1) +
  theme(aspect.ratio = 1) + labs(x = NULL, y = NULL)
bar + coord_flip()
bar + coord_polar()
```





The layered grammar of graphics

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION>
```

- A formal system for building plots,
- Uniquely describes *any* plot as a combination of
  - a dataset,
  - a geom,
  - a set of mappings,
  - a stat,
  - a position adjustment,
  - a coordinate system,
  - and a faceting scheme.

Example

1. Begin with the **`diamonds`** data set

2. Compute counts for each cut value with **`stat_count()`**.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

Example

3. Represent each observation with a bar.
4. Map the **`fill`** of each bar to the **`..count..`** variable.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

Example

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

5. Place geoms in a cartesian coordinate system.

6. Map the y values to **..count..** and the x values to **cut**.